

در این آزمایش، با فرایند کامپایل آشنا می شویم و بر کد اسمبلی و C مروری خواهیم داشت.

خوب است کمی با رسپیری پای آشنا شویم:

رزبری پای یک رایانه کوچک و تک بردی (Single Board Computer) است که توسط بنیاد رزبری پای (Raspberry Pi Foundation) ساخته شده است. این رایانه نسبتاً کوچک اما قدرتمند در اختیار کاربران قرار می دهد. رزبری پای از پردازنده ARM برخوردار است و از سیستم عامل لینوکس پشتیبانی می کند.

ویژگی ها و کاربردهای رزبری پای شامل موارد زیر است:

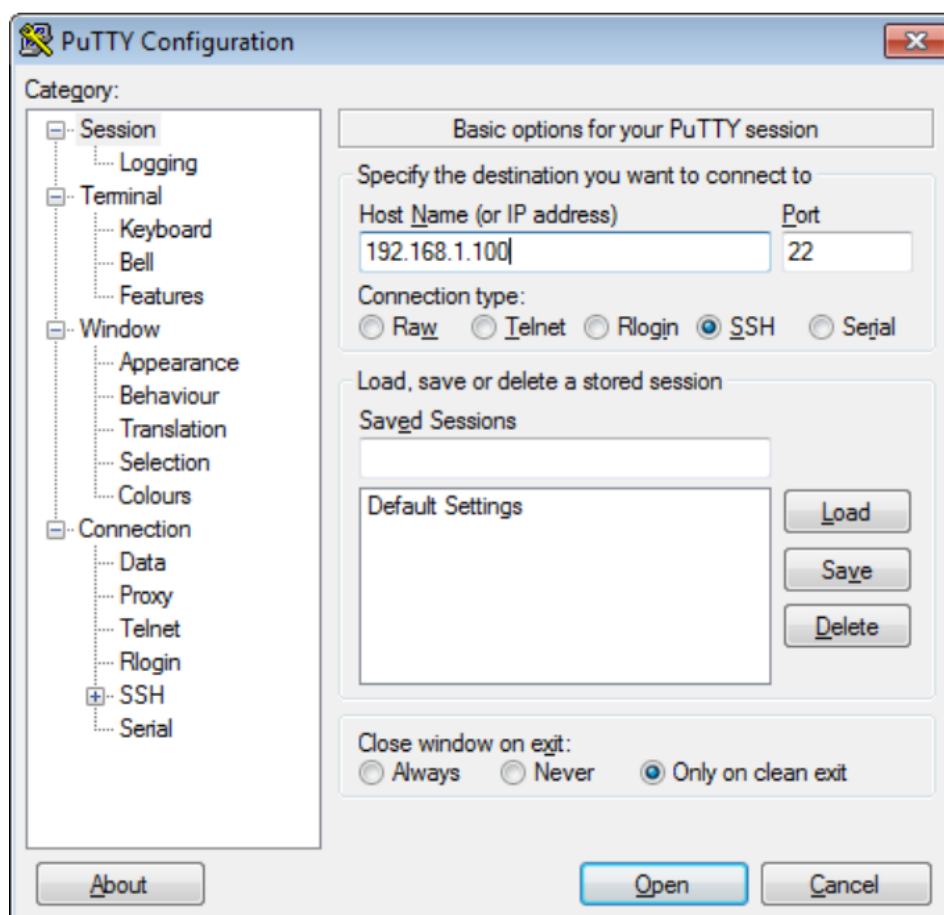
۱. آموزش علوم کامپیوتر: رزبری پای با کوچک و قیمت ارزان خود، یک ابزار عالی برای آموزش علوم کامپیوتر، برنامه نویسی و الکترونیک می باشد. آموزش کودکان و نوجوانان را در زمینه برنامه نویسی راحتتر می کند.
۲. پروژه های الکترونیکی: با استفاده از رزبری پای و اتصال آن به ماژول های الکترونیکی، می توانید پروژه های الکترونیکی متنوعی را مانند رباتیک، سیستم خانه هوشمند، سامانه خودرو و غیره را پیاده سازی کنید.
۳. سرور خانگی: رزبری پای می تواند به عنوان یک سرور خانگی برای ذخیره و مدیریت فایل ها، اشتراک گذاری رسانه، سامانه اطلاعات آب و هوا و سایر خدمات شبکه استفاده شود.
۴. اینترنت اشیا (IoT): با استفاده از رزبری پای، می توانید به راحتی به اینترنت اشیا و دنیای متصل به شبکه متصل شوید. می توانید سنسورها و دستگاه های مختلف را به آن متصل کنید و از آن برای کنترل و نظارت بر این دستگاه ها استفاده کنید.
۵. پروژه های صوتی و تصویری: رزبری پای همچنین قابلیت پخش ویدیو، صدا، وبگردی، پخش پادکست و موارد دیگر را نیز داراست. می توانید از آن برای راه اندازی سیستم های صوتی خانگی، سامانه پخش فیلم و سیستم های نمایش تعاملی استفاده کنید.

برای راه اندازی رسپیری پای ۳ مرحله را طی کردیم:

- ۱- چک کردیم که micro sd حتماً روی آن قرار داشته باشد
 - ۲- کابل USB را برای تغذیه برد قرار دادیم.
 - ۳- کابل اترانت را به جهت ارتباط با برد وصل کردیم.
- طبق شرح گفته شده، micro SD را در ابتدا فرمت کردیم.
- ادرس IP که روی این micro SD بود مانند زیر بود:

```
address 192.168.1.100
netmask 255.255.255.0
gateway 192.168.1.1
```

حال می‌خواهیم ssh را با raspberry pi متصل کنیم. برای این کار نرم افزار putty را بر روی سیستم‌مان اجرا نمودیم. و در قسمت Host Name آدرس IP را وارد کردیم که به خطا خورد و متوجه شدیم باید آدرس IP را خودمان از راه دیگری بدست آوریم.



البته وارد کردن ssh نیز دستخوش چالش هایی بود از جمله اینکه آدرس درست رزبری پای روی *local network* را ابتدا پیدا کردیم و در جای درست وارد نمودیم. برای این منظور، دستور "*ping raspberrypi-local*" را وارد کردیم. این دستور در صورتی که *Raspberry Pi* قابل دسترسی باشد، *IP address* آن را نمایش می دهد. و حال ما *IP address* را وارد کردیم.

گاهی سیستم به ما ارور *ping request could not find host raspberrypi-local. please check the name and try again* را می داد. که با چک کردن برخی اتصالات و کابل ها این ارور نیز رفع شد.

```

C:\Users\mohammad>ssh pi@169.254.52.250
pi@169.254.52.250's password:
Linux raspberrypi 4.1.13-v7+ #826 SMP PREEMPT Fri Nov 13 20:19:03 GMT 2015 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Oct 20 10:51:28 2015 from nima-lenovo.local

NOTICE: the software on this Raspberry Pi has not been fully configured. Please run 'sudo raspi-config'

pi@raspberrypi ~ $ exit

```

بخش اول:

در این مرحله، فایل `hello_world.c` را ایجاد کردیم و برنامه ای نوشتیم که `Hello, World` را چاپ کند و با کامپایلر `gcc` آن را کامپایل کردیم.

```

pi@raspberrypi ~ $ gcc hello_world.c
pi@raspberrypi ~ $ ./a.out
Hello world
pi@raspberrypi ~ $ client_loop: send disconnect: Connection reset

C:\Users\mohammad>ssh pi@169.254.52.250
pi@169.254.52.250's password:
Linux raspberrypi 4.1.13-v7+ #826 SMP PREEMPT Fri Nov 13 20:19:03 GMT 2015 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Oct 20 10:19:09 2015 from desktop-eherdkj.local

NOTICE: the software on this Raspberry Pi has not been fully configured. Please run 'sudo raspi-config'

pi@raspberrypi ~ $ cat hello_world.c
#include <stdio.h>
int main()
{
    printf("Hello world\n");
    return 0;
}

pi@raspberrypi ~ $

```

`Cat` دستوری است که محتوای داخل یک فایل را میخواند و از این دستور در بالا استفاده نمودیم.

البته می توان به کمک `touch` و `mkdir` نیز فایل و پوشه های مورد نظرم را بسازیم.

۱. دستور: "mkdir"

○ این دستور به شما اجازه می دهد یک دایرکتوری جدید ایجاد کنید.

Syntax: `mkdir <directory_name>` ○

○ به عنوان مثال، اگر بخواهید یک دایرکتوری با نام "documents" بسازید، می‌توانید دستور زیر را وارد کنید `mkdir` :
 documents بعد از اجرای این دستور، دایرکتوری با نام "documents" در مسیر کنونی ایجاد خواهد شد.

۲. دستور: "touch"

○ این دستور برای ایجاد یک فایل جدید استفاده می‌شود یا زمان آخرین دسترسی به فایل را به روز می‌کند.

Syntax: `touch <file_name>` ○

○ به عنوان مثال، اگر بخواهید یک فایل با نام "example.txt" ایجاد کنید، می‌توانید دستور زیر را وارد کنید `touch` :
 example.txt در صورتی که فایل با نام "example.txt" وجود نداشته باشد، این دستور آن را ایجاد می‌کند. در صورتی که فایل وجود داشته باشد، زمان آخرین دسترسی به آن به روز می‌شود.

با استفاده از gcc کامپایل میکنیم

خروجی فایل:

```
pi@raspberrypi ~ $ gcc hello_world.c
pi@raspberrypi ~ $ ./a.out
Hello world
pi@raspberrypi ~ $
```

بخش دوم:

فایل `type_test.c` را ایجاد کردیم و برنامه ای را نوشتیم که ۲۰ کاراکتر از `a` تا `z` به صورت رندم تولید کند و دقت و سرعت کاربر در وارد کردن آن رشته را بسنجد و توسط `gcc` آن را کامپایل کردیم.

```

pi@raspberrypi ~ $ cat type_test.c
# include <stdio.h>
# include <time.h>
# include <stdlib.h>

float find_mistake(char inp[21], char out[21]){
    int i;
    int mistake=0;
    for( i=0 ; i<20; i++){
        if(inp[i] != out[i])
            mistake++;
    }
    return (20 - mistake) * 5;
}

int main(){
    char sentence[21];
    char random_list[21];
    int seed ;
    seed = 0;
    srand (time(seed));
    int i;
    for( i=0; i<20; ++i)
    {
        random_list[i] = (char) ('a' + (rand()%26));
    }
    random_list[20] = '\0' ;
    printf("%s\nEnter the string : \n", random_list);
    time_t start_time = time(0);
    scanf("%s", &sentence);

```

```

    int i;
    int mistake=0;
    for( i=0 ; i<20; i++){
        if(inp[i] != out[i])
            mistake++;
    }
    return (20 - mistake) * 5;
}

int main(){
    char sentence[21];
    char random_list[21];
    int seed ;
    seed = 0;
    srand (time(seed));
    int i;
    for( i=0; i<20; ++i)
    {
        random_list[i] = (char) ('a' + (rand()%26));
    }
    random_list[20] = '\0' ;
    printf("%s\nEnter the string : \n", random_list);
    time_t start_time = time(0);
    scanf("%s", &sentence);
    time_t end_time = time(0);
    printf("Timing = %d\n - acc = %f\n", end_time-start_time, find_mistake(sentence, random_list));
    return 0;
}

pi@raspberrypi ~ $

```

```

int main(){
    char sentence[21];
    char random_list[21];
    int seed ;
    seed = 0;
    srand (time(seed));
    int i;
    for( i=0; i<20; ++i)
    {
        random_list[i] = (char) ('a' + (rand()%26));
    }
    random_list[20] = '\0' ;
    printf("%s\nEnter the string : \n", random_list);
    time_t start_time = time(0);
    scanf("%s", &sentence);
    time_t end_time = time(0);
    printf("Timing = %d\n - acc = %f\n", end_time-start_time, find_mistake(sentence, random_list));
    return 0;
}

```

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
 ^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

pi@raspberrypi ~ $ pi@raspberrypi ~ $ nano type_test.c
pi@raspberrypi ~ $ gcc type_test.c
type_test.c: In function 'find_mistake':
type_test.c:7:2: error: 'for' loop initial declarations are only allowed in C99 mode
type_test.c:7:2: note: use option -std=c99 or -std=gnu99 to compile your code
type_test.c: In function 'main':
type_test.c:19:2: warning: passing argument 1 of 'time' makes pointer from integer without a cast [enabled by default]
/usr/include/time.h:186:15: note: expected 'time_t *' but argument is of type 'int'
type_test.c:20:2: error: 'for' loop initial declarations are only allowed in C99 mode
pi@raspberrypi ~ $ nano type_test.c
pi@raspberrypi ~ $ pi@raspberrypi ~ $ gcc type_test.c
type_test.c: In function 'main':
type_test.c:20:2: warning: passing argument 1 of 'time' makes pointer from integer without a cast [enabled by default]
/usr/include/time.h:186:15: note: expected 'time_t *' but argument is of type 'int'
pi@raspberrypi ~ $ ./a.out
pyhrzenwwljeprrwilvl
Enter the string :
pyhrzenwwljeprrwilvl
Timing = 23
- acc = 100.000000

```

